

WHAT IS CLAIMED IS:

1. A method for writing predetermined data to a flash memory,
the method comprising the steps of:

(a) receiving a request to write the predetermined data to a page to
which data has been written;

5 (b) writing the predetermined data to a log block corresponding to a
data block containing the page;

(c) receiving a request to write ~~the~~ predetermined data to the page
again; and

(d) writing ~~the~~ predetermined data to an empty free page in the log
10 block.

2. The method of claim 1, wherein the step (b) comprises the step
(b11) of writing the predetermined data to an empty free page.

3. The method of claim 1, wherein the step (b) comprises the steps
of:

(b21) allocating the log block; and

(b22) writing the predetermined data to an empty page at the same
5 position as the requested page in the data block.

4. A method for writing predetermined data to a flash memory,
the method comprising the steps of:

10029966-133101

- (a) receiving a request to write the predetermined data to a page;
- (b) allocating a log block 1-1 corresponding to a first data block
- 5 containing the page;
- (c) writing the predetermined data to an empty page in the log block 1-1;
- (d) receiving a request to write the predetermined data to the page again; and
- 10 (e) writing the predetermined data to an empty free page in the log block 1-1.

5. The method of claim 4, wherein the step (b) comprises the steps of:

- (b1) performing a block merge to create a third data block based on a second data block and a second log block corresponding to the second data
- 5 block; and
- (b2) allocating a free block obtained by performing an erase operation on the second data block as the log block 1-1.

6. The method of claim 5, wherein the step (b1) is performed when a free block to be allocated as the log block 1-1 does not exist.

7. The method of claim 5, wherein the step (b1) is performed when all pages of the existing log block corresponding to the first data block have been used.

8. The method of claim 5, wherein the step (b1) comprises the step of (b11) performing a switch merge to change the second log block to the third data block when pages of the second log block are arranged in the same order that pages of the second data block are arranged, and the pages of the second log block correspond one-to-one to the pages of the second data block.

9. The method of claim 5, wherein the step (b1) comprises the step of (b12) performing a copy merge to copy corresponding pages of the second data block to free pages in the second log block and create the third data block when the pages in the second log block are requested to be written only once.

10. The method of claim 5, wherein the step (b1) comprises the step of (b13) performing a simple merge to copy the latest pages in the second log block to free pages of a free block to which data has not been written and copy a corresponding page of the second data block to the remaining free pages thereof, thereby creating the third data block.

11. The method of claim 4, wherein the step (e) comprises the steps of:

(e1) allocating a new log block 1-2 if a free page does not exist in the log block 1-1; and

(e2) writing the predetermined data to a free page in the log block 1-2.

12. The method of claim 11, wherein the step (e1) comprises the

steps of:

(e11) performing a switch merge to change the log block to a second data block when pages of the log block 1-1 are arranged in the order in which
5 pages of the first data block are arranged and the pages of the log block 1-1 correspond one-to-one to the pages of the first data block, and

(e12) allocating a free block obtained by performing an erase operation on the first data block as the log block 1-2.

13. The method of claim 11, wherein the step (e1) comprises the steps of:

(e21) performing a copy merge to copy corresponding pages in the first data block to a free page in the log block 1-1 when pages in the
5 log block 1-1 are requested to be written only once; and

(e22) allocating a free block obtained by performing an erase operation on the first data block as the log block 1-2.

14. The method of claim 11, wherein the step (e1) comprises the steps of:

(e31) performing a simple merge to copy the latest pages in the log block 1-1 to free pages of a free block and copy a corresponding page of the
5 first data block to the remaining free pages thereof, thereby creating a second data block; and

(e32) allocating a free block obtained by performing an erase operation on the first data block or the log block 1-1 as the log block 1-2.

15. The method of claim 11, wherein the step (e2) comprises the step of (e21) writing the predetermined data to a free page at the same position as the requested page in the data block.

16. A method for reading predetermined data from a flash memory, the method comprising the steps of:

(a) searching a log pointer table for an entry in which a block address portion of a logical address of a requested page is recorded;

5 (b) checking whether the logical address of the requested page exists in the found entry; and

(c) referring to a physical address of a corresponding log block recorded in the found entry and a position at which the logical address of the requested page is written to the found entry and accessing a corresponding
10 page of the log block.

17. The method of claim 16, wherein, in the step (c), the corresponding page in the log block is accessed at the same position as the position to which the logical address of the requested page is written to the found entry.

18. A method for managing a flash memory including a data block and a log block for writing data for updating the data block, the method

Claims 18-38
OK for
7/14/6
JH

comprising the steps of:

(a) when pages of a first data block are arranged in the same order in
5 which pages of a first log block corresponding to the first data block are
arranged and all the pages of the first data block map one-to-one with the
pages of the first log block, changing the first log block to a second data block;
and

(b) updating address conversion information.

19. The method of claim 18, prior to the step (a), further
comprising the step of (a0) writing recovery information for recovering data in
the event of a system failure during the step (a) or (b).

20. The method of claim 19, further comprising the step of (c)
recovering data referring to the recovery information in the event of a system
failure during the step (a) or (b).

21. The method of claim 20, wherein the recovery information
includes a list of free blocks, a list of log blocks, and a log pointer table which
is the data structure for managing the log blocks.

22. The method of claim 21, wherein the log pointer table contains
log pointer table entries corresponding one-to-one to the log blocks, each entry
mapping a physical address of a log block to a logical address of a
corresponding data block and storing logical addresses of requested pages of a
5 data block in the order in which pages of a corresponding log block are

physically arranged.

23. The method of claim 21, wherein the log pointer table is constructed by scanning a log block region in which the log blocks are stored.

24. A method for managing a flash memory including a data block and a log block for writing data for updating the data block, the method comprising the steps of:

(a) when pages in a first log block are requested to be written only once, copying a corresponding page of a first data block to a free page of the first log block in order to create a second data block; and

(b) updating address conversion information.

25. The method of claim 24, prior to the step (a), further comprising the step of (a0) writing recovery information for recovering data in the event of a system failure during the step (a) or (b).

26. The method of claim 25, further comprising the step (c) recovering data referring to the recovery information in the event of a system failure during the step (a) or (b).

27. The method of claim 26, wherein the recovery information includes a list of free blocks, a list of log blocks, and a log pointer table which is the data structure for managing the log blocks.

28. The method of claim 27, wherein the log pointer table contains log pointer table entries corresponding one-to-one to the log blocks, each entry

mapping a physical address of a log block to a logical address of a
corresponding data block and storing logical addresses of requested pages of a
5 data block in the order in which pages of a corresponding log block are
physically arranged.

29. The method of claim 27, wherein the log pointer table is
constructed by scanning a log block region in which the log blocks are stored.

30. A method for managing a flash memory including a data block
and a log block for writing data for updating the data block, the method
comprising the steps of:

(a) copying the latest pages in a first log block to a free block to which
5 data has not been written and copying a corresponding page of a first data
block corresponding to the first log block to a remaining free page to create a
second data block; and

(b) updating address conversion information.

31. The method of claim 30, prior to the step (a), further
comprising the step of (a0) writing recovery information for recovering data in
the event of a system failure during the step (a) or (b).

32. The method of claim 31, further comprising the step (c) of
recovering data referring to the recovery information in the event of a system
failure during the step (a) or (b).

33. The method of claim 32, wherein the recovery information

includes a list of free blocks, a list of log blocks, and a log pointer table which is a data structure for managing the log blocks.

34. The method of claim 33, wherein the log pointer table contains log pointer table entries corresponding one-to-one to the log blocks, each entry mapping a physical address of a log block to a logical address of a corresponding data block and storing logical addresses of requested pages of a data block in the order in which pages of a corresponding log block are physically arranged.

35. The method of claim 34, wherein the log pointer table is constructed by scanning a log block region in which the log blocks are stored.

36. A method for managing a flash memory including a data block and a log block for writing data for updating the data block, the method comprising the steps of:

(a) allocating a predetermined region to a flash memory and writing lists of data blocks and log blocks and a data structure for managing the log blocks to the predetermined region as recovery information;

(b) checking states currently being written to the flash memory based on the recovery information in the event of a system failure to determine whether an error occurs; and

(c) if the error occurs, recovering data based on the recovery information.

37. The method of claim 36, wherein the recovery information includes a list of free blocks.

38. The method of claim 37, wherein the recovery information further includes a list of log blocks and a log pointer table which is the data structure for managing the log blocks and wherein the log pointer table contains log pointer table entries corresponding to the number of log blocks,
5 each entry mapping a physical address of a log block to a logical address of a corresponding data block and storing logical addresses of requested pages of a data block in the order in which pages of a corresponding log block are physically arranged.

1002996-123101
FOIEAT 99662001